Tight Approximation Ratio of a General Greedy Splitting Algorithm for the Minimum k-Way Cut Problem

MINGYU XIAO   LEIZHEN CAI   ANDREW C. YAO

Department of Computer Science and Engineering
The Chinese University of Hong Kong
Hong Kong SAR, CHINA

Email: myxiao(lcai)@cse.cuhk.edu.hk, andrewcyao@tsinghua.edu.cn

Abstract

For an edge-weighted connected undirected graph, the minimum k-way cut problem is to find a subset of edges of minimum total weight whose removal separates the graph into k connected components. The problem is NP-hard when k is part of the input and W[1]-hard when k is taken as a parameter.

A simple algorithm for approximating a minimum k-way cut is to iteratively increase the number of components of the graph by $h - 1$, where $2 \le h \le k$, until the graph has k components. The approximation ratio of this algorithm is known for $h \le 3$ but is open for $h \ge 4$.

In this paper, we consider a general algorithm that iteratively increases the number of components of the graph by $h_i - 1$, where $h_1 \le h_2 \le \cdots \le h_q$ and $\sum_{i=1}^{q}(h_i - 1) = k - 1$. We prove that the approximation ratio of this general algorithm is $2 - (\sum_{i=1}^{q}\binom{h_i}{2})/\binom{k}{2}$, which is tight. Our result implies that the approximation ratio of the simple algorithm is $2 - h/k + O(h^2/k^2)$ in general and $2 - h/k$ if $k - 1$ is a multiple of $h - 1$.

Key words approximation algorithm, k-way cut, k-way split.

1 Introduction

Let $G = (V, E; w)$ a connected undirected graph with n vertices and m edges, where each edge e has a positive weight $w(e)$, and k a positive integer. A *k-way cut* of G is a subset of edges whose removal separates the graph into k connected components, and the *minimum k-way cut* problem is to find a k-way cut of minimum total weight. We note that k-way cuts are also referred to as *k-cuts* or *multi-component cuts* in the literature.

The minimum k-way cut problem is a natural generalization of the classical *minimum cut* problem and has been very well studied in the literature. Goldschmidt and Hochbaum [1] proved that the minimum k-way cut problem is NP-hard when k is part of the input and gave an $O(n^{(1/2-o(1))k^2})$ algorithm. Kamidoi et al. [2] presented an $O(n^{4k/(1-1.71/\sqrt{k})-31})$ algorithm, and Xiao [3] presented an $O(n^{4k-\log k})$ algorithm. These three algorithms are based on a divide-and-conquer method. Karger and Stein [4] proposed a randomized algorithm that runs in $O(n^{2k-2} \log^3 n)$ expected time. Recently, Thorup [5] obtained an $O(n^{2k} \log n)$ algorithm via tree packing. On the other hand, Downey et al. [6] showed that the problem is W[1]-hard when k is taken as a parameter, which indicates that it is very unlikely to solve the problem in $f(k)n^{O(1)}$ time for any function $f(k)$. We also note that faster algorithms are available for small k. Nagamochi and Ibaraki [7], and Hao and Orlin [8] solved the minimum 2-way cut problem (i.e., the minimum cut problem) in $O(mn + n^2 \log n)$ and $O(mn \log(n^2/m))$ time respectively. Burlet and Goldschmidt [9] solved the minimum 3-way cut problem in $\widetilde{O}(mn^3)$ time, Nagamochi and Ibaraki [10] gave $\widetilde{O}(mn^k)$ algorithms for $k \leq 4$, and Nagamochi et al. [11] extended this result for $k \leq 6$. Furthermore, Levine [12] obtained $O(mn^{k-2} \log^3 n)$ randomized algorithms for $k \leq 6$.

In terms of approximation algorithms, Saran and Vazirani [13] gave two simple algorithms of approximation ratio $2 - 2/k$. Naor and Rabani [14] obtained an integer program formulation of this problem with integrality gap 2, and Ravi and Sinha [15] also derived a 2-approximation algorithm via the network strength method.

A simple algorithm [13] for approximating a minimum k-way cut is to iteratively increase the number of components of the graph by $h-1$, where $2 \leq h \leq k$, until the graph has k components. This algorithm has an approximation ratio of $2 - 2/k$ for $h = 2$ [13], and Kapoor [16] claimed that it achieves ratio $2 - \alpha(h, k)$ for $h \geq 3$, where $\alpha(h, k) = h/k - (h-2)/k^2 + O(h/k^3)$. Unfortunately, his proof for $h \geq 3$ is incomplete. Later, Zhao et al. [17] established Kapoor's claim for $h = 3$: the ratio is $2 - 3/k$ for odd k and $2 - (3k-4)/(k^2 - k)$ for even k. However, for $h \geq 4$, it seems quite difficult to analyze the performance of this algorithm and it has been an open problem whether we get a better approximation ratio with this approach.

In this paper, we consider a general algorithm that iteratively increases the number of components of the graph by $h_i - 1$, where $h_1 \leq h_2 \leq \cdots \leq h_q$ and $\sum_{i=1}^{q}(h_i - 1) = k - 1$. We prove that the approximation ratio of this general

algorithm is $2 - (\sum_{i=1}^{q} \binom{h_i}{2})/\binom{k}{2}$, which is tight. Our result implies that the approximation ratio of the simple algorithm is $2 - h/k + O(h^2/k^2)$ in general and $2 - h/k$ if $k - 1$ is a multiple of $h - 1$, which settles the open problem mentioned earlier in the affirmative.

The rest of the paper is organized as follows. In Section 2, we formalize our general greedy splitting algorithms and present our main results on their approximation ratios. We prove our main results in Section 3 while the proof of a purely analytical lemma is given in Section 4, and conclude with some remarks in Section 5.

2   Algorithms and main results

In this section, we formalize our greedy splitting algorithms and present our main results on their approximation ratios. We note that Zhao et al. [18, 19] have studied such algorithms for general multiway cut and partition problems. First we extend the notion of k-way cuts to disconnected graphs. A *k-way split* of a graph is a subset of edges whose removal increases the number of components by $k - 1$. Therefore for a connected graph, a k-way split is equivalent to a k-way cut. We note that the time for finding a minimum k-way split in a general graph is the same as finding a k-way cut [17].

One general approach for finding a light k-way cut is to find minimum h_i-way splits successively for a given sequence $(h_1, h_2, \cdots, h_q)$.

Algorithm iterative-split$(G, k, (h_1, h_2, \cdots, h_q))$

Input: Connected graph $G = (V, E; w)$, integer k and sequence $(h_1, h_2, \cdots, h_q)$ of integers satisfying $2 \leq h_1 \leq h_2 \leq \cdots \leq h_q$ and $\sum_{i=1}^{q}(h_i - 1) = k - 1$.

Output: A k-way cut of G.

1. For $i := 1$ to q find a minimum h_i-way split C_i of G and let $G \leftarrow G - C_i$.

2. Return $\bigcup_{i=0}^{q} C_i$ as a k-way cut.

A special case of the above algorithm is when all h_i's in the integer sequence, with the possible exception of the first one, are equal. The following gives a precise description of this special case.

Algorithm iterative-h-split(G, k, h)

Input: Connected graph $G = (V, E; w)$, integers k and h.

Output: A k-way cut of G.

1. Let $p = \lfloor \frac{k-1}{h-1} \rfloor$ and $r = (k - 1) \bmod (h - 1)$.

2. If $r \neq 0$, then find a minimum $(r+1)-$way split C_0 of G and let $G \leftarrow G - C_0$.

3. For $i := 1$ to p find a minimum h-way split C_i of G and let $G \leftarrow G - C_i$.

4. Return $\bigcup_{i=0}^{p} C_i$ as a k-way cut.

The above two algorithms run in polynomial time if h_q and h are bounded by some constant, and our main results of the paper are the following two tight bounds for their approximation ratios.

Theorem 2.1 *The approximation ratio of algorithm* **iterative-split** *is*

$$2 - \frac{\sum_{i=1}^{q} \binom{h_i}{2}}{\binom{k}{2}}.$$

Corollary 2.2 *The approximation ratio of algorithm* **iterative-h-split** *is*

$$2 - \frac{h}{k} + \frac{(h-1-r)r}{k(k-1)} = 2 - \frac{h}{k} + O(\frac{h^2}{k^2}),$$

where $r = (k-1) \mod (h-1)$.

Remark. We note that when $k-1$ is a multiple of $h-1$, **iterative-h-split** is a $(2 - h/k)$-approximation algorithm, and Corollary 2.2 for $h = 3$ yields a result of Zhao et al. [17].

3   Performance analysis

In this section, we will prove our main results on the approximation ratios of our approximation algorithms. For this purpose, we first establish a relation between the weight $w(C_h)$ of a minimum h-way split C_h and the weight $w(C_k)$ of a k-way split C_k, which will be the main tool in our analysis. For convenience, we allow $h = 1$ (note that a minimum 1-way split is an empty set). For a collection of mutually disjoint subsets $V_1, V_2, \cdots, V_t \in V$, we use $[V_1, V_2, \cdots, V_t]$ to denote the set of edges uv such that $u \in V_i$ and $v \in V_j$ for some $V_i \neq V_j$.

Lemma 3.1 *Let G be an edge-weighted graph, $h \geq 1$, and $k \geq \max\{2, h\}$. For any minimum h-way split C_h and any k-way split C_k of G, the following holds.*

$$\frac{w(C_h)}{w(C_k)} \leq (2 - \frac{h}{k})\frac{h-1}{k-1}. \tag{1}$$

Proof. First we consider the case that G is connected. In this case, C_k and C_h, respectively, are k-way and minimum h-way cuts of G, and thus C_k corresponds to a partition $\Pi = \{V_1, V_2, \ldots, V_k\}$ of the vertex set V of G such that each V_i is a component of $G - C_k$.

We can merge any $k - (h-1)$ elements in Π into one element to form a new partition $\Pi' = \{V_1', V_2', \ldots, V_h'\}$ of V. Let $E(\Pi') = [V_1', V_2', \ldots, V_h']$. Then $G - E(\Pi')$ has at least h components, and therefore the weight $w(E(\Pi'))$ of $E(\Pi')$ is at least $w(C_h)$. There are $\binom{k}{h-1}$ different ways to form Π', and therefore the total weight W of all $E(\Pi')$ is at least $\binom{k}{h-1}w(C_h)$.

On the other hand, we can put an upper bound on W by relating it to the weight of C_k. Consider the set E_{ij} of edges in C_k between V_i and V_j. For a partition Π', $E_{ij} \subseteq E(\Pi')$ iff V_i and V_j are not merged in forming Π'. The number of Π's for which V_i and V_j are merged is $\binom{k-2}{h-1}$, implying that each E_{ij} is counted $\binom{k}{h-1} - \binom{k-2}{h-1}$ times in calculating W. Therefore

$$W = (\binom{k}{h-1} - \binom{k-2}{h-1}) \cdot w(C_k) \geq \binom{k}{h-1} \cdot w(C_h),$$

which yields the inequality in the lemma.

For the case that G is disconnected, we construct a connected graph $G' = (V', E'; w')$ from G as follows:

1. Add a new vertex v.

2. For each component H of G, add an edge e_H between v and an arbitrary vertex of H.

3. Set the weight of e_H to ∞.

4. Set $w'(e) = w(e)$ for all other edges of G'.

Then every k-way split in G is a k-way cut in G', and every minimum h-way split in G is a minimum h-way cut in G'. Since G' is connected, the lemma holds for G' and hence for k-way and minimum h-way splits of G. ∎

For convenience, define for all $h \geq 1$ and $k \geq \max\{2, h\}$,

$$f(k, h) = (2 - \frac{h}{k})\frac{h-1}{k-1}.$$

We note that the bound in Lemma 3.1 is tight, which can be seen by considering a k-way cut and a minimum h-way cut of the complete graph K_k. This also gives a combinatorial explanation of $f(k, h)$: the ratio between the number of edges covered by $h - 1$ vertices in K_k and the number of edges of K_k. We also need the following properties of $f(k, h)$ in our analysis.

Fact 3.2 *Function $f(k, h)$ monotonically increases for $h \in [1, k]$ and monotonically decreases for $k \in [h, \infty)$.*

Fact 3.3 *For all $a \geq 0, h \geq 2$, and $k \geq a + h$,*

$$f(k - a, h)(1 - f(k, a + 1)) \leq f(k, h). \tag{2}$$

Proof. Straightforward manipulation gives

$$f(k - a, h)(1 - f(k, a + 1)) = (2 - \frac{2a + h}{k})\frac{h-1}{k-1} \leq f(k, h).$$

■

The next inequality is an analytical result critical to the proof of our main theorem. Let $q \geq 2$. For any integers $2 \leq h_1 \leq h_2 \leq \cdots \leq h_q$, $0 \leq a \leq h_1 - 1$ and $k - 1 \geq \sum_{i=1}^{q}(h_i - 1)$, let

$$D = f(k - a, h_1 - a) + \sum_{i=2}^{q} f(k - a, h_i) \tag{3}$$

and

$$F = \max\{D, \quad f(k, a + 1) + (1 - f(k, a + 1))D\}. \tag{4}$$

Lemma 3.4 $F \leq \sum_{i=1}^{q} f(k, h_i)$.

To avoid distraction from our main discussions, we delay the proof of this purely analytical lemma to Section 4.

We are now ready to prove our main results. For this purpose, we call a sequence $((C_1, h_1), \ldots, (C_q, h_q))$ a *nondecreasing q-sequence of minimum splits* if integers $2 \leq h_1 \leq h_2 \leq \cdots \leq h_q$ and each C_i, $1 \leq i \leq q$, is a minimum h_i-way split of $G_i = G - \bigcup_{j=1}^{i-1} G_j$. To prove Theorem 2.1, it suffices to prove the following theorem. We note that although the proof is an inductive one, the argument in the proof is subtle, and the condition $h_1 \leq h_2 \leq \cdots \leq h_q$ is crucial to the proof.

Theorem 3.5 *Let $((C_1, h_1), \ldots, (C_q, h_q))$ be a nondecreasing q-sequence of minimum splits of a weighted graph $G = (V, E; w)$, where $w : E \to R^+$, and S_k a k-way split of G satisfying $k - 1 \geq \sum_{i=1}^{q}(h_i - 1)$. Then*

$$w(\bigcup_{i=1}^{q} C_i) \leq \sum_{i=1}^{q} f(k, h_i) \cdot w(S_k). \tag{5}$$

Proof. We use induction on q. For $q = 1$, the theorem is established by Lemma 3.1. For the inductive step, let $q \geq 2$, $C_1' = C_1 \cap S_k$, $S_{k'} = S_k - C_1'$, and $C_1'' = C_1 - C_1'$. Then C_1' is an $(a + 1)$-way split of G for some $0 \leq a \leq h_1 - 1$, C_1'' is a minimum $(h_1 - a)$-way split of $G - C_1'$ (otherwise C_1 would not be a minimum h_1-way split of G), and $S_{k'}$ is a $(k - a)$-way split of $G - C_1'$. It follows that $S_{k'}$ is a k'-way split of $G - C_1$ for some $k' \geq k - a$. Note that $((C_2, h_2), \ldots, (C_q, h_q))$ is a nondecreasing $(q - 1)$-sequence of minimum splits of $G - C_1$ and $k' - 1 \geq \sum_{i=2}^{q}(h_i - 1)$. By the induction hypothesis and the fact that each $f(k', h_i)$ is at most $f(k - a, h_i)$ (Fact 3.2), we have

$$w(\bigcup_{i=2}^{q} C_i) \leq \sum_{i=2}^{q} f(k', h_i) \cdot w(S_{k'}) \leq \sum_{i=2}^{q} f(k - a, h_i) \cdot w(S_{k'}). \tag{6}$$

Let $W = w(C_1) + \sum_{i=2}^{q} f(k - a, h_i) \cdot w(S_{k'})$. Then $w(\bigcup_{i=1}^{q} C_i) \leq W$ by (6), and we will establish the theorem by proving $W \leq \sum_{i=1}^{q} f(k, h_i) \cdot w(S_k)$.

If $w(C_1') > f(k, a+1)w(S_k)$, then $w(S_{k'}) = w(S_k) - w(C_1') \leq (1 - f(k, a+1))w(S_k)$. By Lemma 3.1, we have $w(C_1) \leq f(k, h_1) \cdot w(S_k)$ and it follows from Fact 3.3 that

$$
\begin{aligned}
W &\leq (f(k, h_1) + \sum_{i=2}^{q} f(k - a, h_i)(1 - f(k, a+1))) \cdot w(S_k) \\
&\leq \sum_{i=1}^{q} f(k, h_i) \cdot w(S_k).
\end{aligned}
$$

Otherwise, $w(C_1') \leq f(k, a+1) \cdot w(S_k)$ and we have

$$
W = w(C_1') + w(C_1'') + \sum_{i=2}^{q} f(k - a, h_i) \cdot w(S_{k'}).
$$

Since C_1'' is a minimum $(h_1 - a)$-way split of $G - C_1'$, we have $w(C_1'') \leq f(k - a, h_1 - a) \cdot w(S_{k'})$ by Lemma 3.1. It follows that

$$
\begin{aligned}
W &\leq w(C_1') + f(k - a, h_1 - a) \cdot w(S_{k'}) + \sum_{i=2}^{q} f(k - a, h_i) \cdot w(S_{k'}) \\
&= w(C_1') + D \cdot w(S_{k'})
\end{aligned}
$$

for $D = f(k - a, h_1 - a) + \sum_{i=2}^{q} f(k - a, h_i)$ as defined in (3). Define $x = w(C_1')/w(S_k)$ and we have $W \leq (x + (1 - x)D)w(S_k)$. Since $0 \leq x \leq f(k, a+1)$, the maximum value of $x + (1 - x)D$ over the interval $[0, f(k, a+1)]$ must be at either $x = 0$ or $x = f(k, a+1)$ as it is a linear function in x. This means

$$
\frac{W}{w(S_k)} \leq \max\{D, f(k, a+1) + (1 - f(k, a+1))D\}.
$$

Therefore by Lemma 3.4, we have

$$
W \leq (\sum_{i=1}^{q} f(k, h_i)) \cdot w(S_k).
$$

This completes the inductive step and therefore proves the theorem. ∎

We can obtain Theorem 2.1 for Algorithm **iterative-split** from Theorem 3.5 as follows (note that $\sum_{i=1}^{q}(h_i - 1) = k - 1$):

$$
\begin{aligned}
\sum_{i=1}^{q} f(k, h_i) &= \sum_{i=1}^{q}(2 - \frac{h_i}{k})\frac{h_i - 1}{k - 1} \\
&= \frac{2}{k - 1}\sum_{i=1}^{q}(h_i - 1) - \frac{1}{k(k - 1)}\sum_{i=1}^{q} h_i(h_i - 1) \\
&= 2 - \frac{\sum_{i=1}^{q}\binom{h_i}{2}}{\binom{k}{2}}.
\end{aligned}
$$

For Algorithm **iterative-h-split**, we can easily derive Corollary 2.2 from Theorem 2.1.

Remark The bound in Theorem 3.5 is tight for $k - 1 = \sum_{i=1}^{q}(h_i - 1)$ and therefore the approximation ratios in Theorem 2.1 and Corollary 2.2 are tight. To see this, consider the following graph G that consists of the disjoint union of $q + 1$ copies $H_1, H_2, \cdots, H_q, K$ of the complete graph K_k. For each H_i, fix a subset V_i of $h_i - 1$ vertices and let E_i denote edges in H_i that are covered by V_i. Each edge in E_i has weight 1, and each of the remaining edges of H_i has weight ∞. Set the weight of every edge in K to 1.

A minimum k-way split C_k of G consists of all edges in K, but **iterative-split** may return $\bigcup_{i=1}^{q} E_i$ as a k-way split C_k' of G. Since $w(C_k) = \binom{k}{2}$ and $w(C_k') = \sum_{i=1}^{q} |E_i| = f(k, h_i)\binom{k}{2}$, we have $w(C_k')/w(C_k) = \sum_{i=1}^{q} f(k, h_i)$.

4 Proof of Lemma 3.4

In this section, we complete our performance analysis by proving Lemma 3.4: $F \leq \sum_{i=1}^{q} f(k, h_i)$, where $F = \max\{D, \ W'\}$ for $D = f(k-a, h_1-a) + \sum_{i=2}^{q} f(k-a, h_i)$ and $W' = f(k, a+1) + (1 - f(k, a+1))D$. For this purpose, we first derive some useful properties of $f(k, h)$.

Fact 4.1 *For all $h_1, h_2 \geq 0$ and $k \geq \max\{h_1 + h_2 + 1, 2\}$,*

$$f(k, h_1 + h_2 + 1) = f(k, h_1 + 1) + f(k - h_1, h_2 + 1)(1 - f(k, h_1 + 1)).$$

Proof. Let $e(k, h)$ denote the number of edges covered by h vertices in the complete graph K_k, and m_k the number of edges in K_k. Then

$$e(k, h_1 + h_2) = e(k, h_1) + e(k - h_1, h_2),$$

and thus

$$\frac{e(k, h_1 + h_2)}{m_k} = \frac{e(k, h_1)}{m_k} + \frac{e(k - h_1, h_2)}{m_{k-h_1}} \cdot \frac{m_{k-h_1}}{m_k}.$$

Since $m_{k-h_1} = m_k - e(k, h_1)$, we obtain

$$\frac{e(k, h_1 + h_2)}{m_k} = \frac{e(k, h_1)}{m_k} + \frac{e(k - h_1, h_2)}{m_{k-h_1}} \cdot (1 - \frac{e(k, h_1)}{m_k}),$$

and the lemma follows from the fact that $f(k, h) = e(k, h - 1)/m_k$. ∎

Fact 4.2 *For all $a \geq 0, h_2 \geq h_1 \geq 2, k \geq a + h_2$,*

$$f(k - a, h_2) - f(k, h_2) \leq \frac{h_2 - 1}{h_1 - 1}[f(k - a, h_1) - f(k, h_1)].$$

Proof.

$$\Leftrightarrow \quad f(k-a, h_2) - \frac{h_2 - 1}{h_1 - 1} f(k-a, h_1) \leq f(k, h_2) - \frac{h_2 - 1}{h_1 - 1} f(k, h_1)$$

$$\Leftrightarrow \quad -\frac{(h_2 - h_1)(h_2 - 1)}{(k-a)(k-a-1)} \leq -\frac{(h_2 - h_1)(h_2 - 1)}{k(k-1)}$$

$$\Leftrightarrow \quad (k-a)(k-a-1) \leq k(k-1).$$

∎

Fact 4.3 *For all* $a \geq 0, h \geq 2, k \geq a + h,$

$$f(k-a, h-a) + \frac{k-h}{h-1} f(k-a, h) \leq \frac{k-1}{h-1} f(k, h).$$

Proof.

$$\Leftrightarrow \quad \frac{a^2 + a(1 + 2h - 4k) - (h - 2k)(k - 1)}{(k-a)(k-a-1)} \leq \frac{2k - h}{k}$$

$$\Leftrightarrow \quad k(a^2 + a(1 + 2h - 4k) - (h - 2k)(k - 1)) \leq (2k - h)(k-a)(k-a-1)$$

$$\Leftrightarrow \quad a(a+1)(h - k) \leq 0.$$

∎

Fact 4.4 *For all* $2 \leq h_1 \leq h_i$ $(i = 2, 3, \cdots, q), 0 \leq a < h_1, \sum_{i=1}^{q} (h_i - 1) \leq k - 1,$

$$f(k-a, h_1 - a) + \sum_{i=2}^{q} f(k-a, h_i) \leq f(k, h_1) + \sum_{i=2}^{q} f(k, h_i).$$

Proof. Let $\Delta = f(k-a, h_1 - a) + \sum_{i=2}^{q} f(k-a, h_i) - f(k, h_1) - \sum_{i=2}^{q} f(k, h_i).$ By Fact 4.2, we have

$$\sum_{i=2}^{q} (f(k-a, h_i) - f(k, h_i)) \quad \leq \quad \sum_{i=2}^{q} \frac{h_i - 1}{h_1 - 1} (f(k-a, h_1) - f(k, h_1))$$

$$= \quad \frac{k - h_1}{h_1 - 1} (f(k-a, h_1) - f(k, h_1)).$$

Therefore

$$\Delta \quad \leq \quad f(k-a, h_1 - a) - f(k, h_1) + \frac{k - h_1}{h_1 - 1} (f(k-a, h_1) - f(k, h_1))$$

$$= \quad f(k-a, h_1 - a) + \frac{k - h_1}{h_1 - 1} f(k-a, h_1) - \frac{k - 1}{h_1 - 1} f(k, h_1).$$

It follows from Fact 4.3 that $\Delta \leq 0$, which proves the lemma. ∎

Now, we are ready to prove Lemma 3.4: $F \leq \sum_{i=1}^{q} f(k, h_i)$. Recall that $F = \max\{D, W'\}$ for $D = f(k-a, h_1-a) + \sum_{i=2}^{q} f(k-a, h_i)$ and $W' = f(k, a+1) + (1 - f(k, a+1))D$. As $D \leq \sum_{i=1}^{q} f(k, h_i)$ by Fact 4.4, we need only show that $W' \leq \sum_{i=1}^{q} f(k, h_i)$. This can be done by using Fact 4.1 and Fact 3.3 as follows:

$$
\begin{aligned}
W' &= f(k, a+1) - f(k-a, h_1-a)f(k, a+1) + f(k-a, h_1-a) \\
&\quad + \sum_{i=2}^{q} f(k-a, h_i)(1 - f(k, a+1)) \\
&= f(k, h_1) + \sum_{i=2}^{q} f(k-a, h_i)(1 - f(k, a+1)) \quad \text{(by Fact 4.1)} \\
&\leq \sum_{i=1}^{q} f(k, h_i). \quad \text{(by Fact 3.3)}
\end{aligned}
$$

5   Concluding remarks

In this paper, we have determined the exact approximation ratio of a general splitting algorithm **iterative-split** for the minimum k-way cut problem. The answer is a surprisingly simple expression $2 - \sum_{i=1}^{q} \binom{h_i}{2} / \binom{k}{2}$, yet it takes a somewhat subtle and involved inductive argument to prove the result. It would be interesting to find a direct and simpler proof.

We note that for **iterative-split**, the requirement that $h_1 \leq h_2 \leq \cdots \leq h_q$ is crucial for obtaining the approximation ratio of the algorithm, which is unknown if we drop the requirement. We also note that if we restrict h_q to be at most h, then **iterative-h-split**, a special case of **iterative-split**, achieves the best approximation ratio among all possible choices of $h_1 \leq h_2 \leq \cdots \leq h_q$.

Finally, we may use **iterative-split** as a general framework for designing approximation algorithms for various cut and partition problems, and the ideas in this paper may shed light on the analysis of this general approach for these problems.

References

[1] Goldschmidt, O., Hochbaum, D.: A polynomial algorithm for the k-cut problem for fixed k. Mathematics of Operations Research **19**(1) (1994) 24–37 A preliminary version appeared in FOCS1988.

[2] Kamidoi, Y., Yoshida, N., Nagamochi, H.: A deterministic algorithm for finding all minimum k-way cuts. SIAM Journal on Computing **36**(5) (2006) 1329–1341

[3] Xiao, M.: An improved divide-and-conquer algorithm for finding all minimum *k*-way cuts. In: Proceedings of the 19th International Symposium on Algorithms and Computation (ISAAC 2008). (2008) 208–219

[4] Karger, D.R., Stein, C.: A new approach to the minimum cut problem. Journal of the ACM **43**(4) (1996) 601–640 Preliminary portions appeared in SODA1993 and STOC1993.

[5] Thorup, M.: Minimum k-way cuts via deterministic greedy tree packing. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC 2008). (2008) 159–166

[6] Downey, R.G., Estivill-Castro, V., Fellows, M.R., Prieto, E., Rosamond, F.A.: Cutting up is hard to do: the parameterized complexity of k-cut and related problems. Electr. Notes Theor. Comput. Sci. **78** (2003) 1–14

[7] Nagamochi, H., Ibaraki, T.: Computing edge connectivity in multigraphs and capacitated graphs. SIAM Journal on Discrete Mathematics **5**(1) (1992) 54–66

[8] Hao, J., Orlin, J.B.: A faster algorithm for finding the minimum cut in a graph. In: Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms (SODA1992), Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (1992) 165–174

[9] Burlet, M., Goldschmidt, O.: A new and improved algorithm for the 3-cut problem. Operations Research Letters **21**(5) (1997) 225–227

[10] Nagamochi, H., Ibaraki, T.: A fast algorithm for computing minimum 3-way and 4-way cuts. Mathematical Programming **88**(3) (2000) 507–520

[11] Nagamochi, H., Katayama, S., Ibaraki, T.: A faster algorithm for computing minimum 5-way and 6-way cuts in graphs. In: Asano, T., Imai, H., Lee, D.T., Nakano, S.-i., Tokuyama, T. (eds.) COCOON 1999. LNCS, vol. 1627, Springer, Heidelberg. (1999)

[12] Levine, M.S.: Fast randomized algorithms for computing minimum {3,4,5,6}-way cuts. In: Proceedings of the 11th annual ACM-SIAM symposium on Discrete algorithms (SODA2000), Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2000) 735–742

[13] Saran, H., Vazirani, V.V.: Finding k-cuts within twice the optimal. SIAM J. Comput. **24**(1) (1995) 101–108 A preliminary version appeared in FOCS1991.

[14] Naor, J., Rabani, Y.: Tree packing and approximating k-cuts. In: Proceedings of the twelfth annual ACM-SIAM symposium on discrete algorithms (SODA2001), Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2001) 26–27

[15] Ravi, R., Sinha, A.: Approximating k-cuts via network strength. In: Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms (SODA2002), Philadelphia, PA, USA, Society for Industrial and Applied Mathematics (2002) 621–622

[16] Kapoor, S.: On minimum 3-cuts and approximating k-cuts using cut trees. In: Proceedings of the 5th International IPCO Conference on Integer Programming and Combinatorial Optimization, London, UK, Springer-Verlag (1996) 132–146

[17] Zhao, L., Nagamochi, H., Ibaraki, T.: Approximating the minimum k-way cut in a graph via minimum 3-way cuts. J. Comb. Optim. **5**(4) (2001) 397–410 A preliminary version appeared in ISAAC1999.

[18] Zhao, L., Nagamochi, H., Ibaraki, T.: Greedy splitting algorithms for approximating multiway partition problems. Math. Program. **102**(1) (2005) 167–183

[19] Zhao, L., Nagamochi, H., Ibaraki, T.: On generalized greedy splitting algorithms for multiway partition problems. Discrete Applied Mathematics **143**(1-3) (2004) 130–143